Explainable Activity Recognition over Interpretable Models

Claudio Bettini
Dept. of Computer Science
University of Milan
Milan, Italy
claudio.bettini@unimi.it

Gabriele Civitarese
Dept. of Computer Science
University of Milan
Milan, Italy
gabriele.civitarese@unimi.it

Michele Fiori
Dept. of Computer Science
University of Milan
Milan, Italy
michele.fiori@studenti.unimi.it

***Abstract*—The majority of the approaches to sensor-based activity recognition are based on supervised machine learning. While these methods reach high recognition rates, a major challenge is to understand the rationale behind the predictions of the classifier. Indeed, those predictions may have a relevant impact on the follow-up actions taken in a smart living environment. We propose a novel approach for eXplainable Activity Recognition (XAR) based on interpretable machine learning models. We generate explanations by combining the feature values with the feature importance obtained from the underlying trained classifier. A quantitative evaluation on a real dataset of ADLs shows that our method is effective in providing explanations consistent with common knowledge. By comparing two popular ML models, our results also show that *one versus one* classifiers can provide better explanations in our framework.**

***Index Terms*—activity recognition, explainable artificial intelligence, smart-homes**

I. Introduction

The recognition of Activities of Daily Living (ADL) is a well-known research area that has been widely explored in the last years [1]. Recognizing the high-level activities that humans perform in their daily life is crucial for several applications, including ambient assisted living, well being, and health-care [2]. ADLs recognition through wearable and environmental sensors is mainly based on supervised machine learning approaches [3]. These approaches can reach significantly high recognition rates, when large labeled training sets are available. However, one of the major drawbacks of machine learning methods in general is that the classifiers do not provide an explanation for their output. A lot of recent research efforts in AI have been focused on eXplainable Artificial Intelligence (XAI) [4]. XAI aims at building machine learning models capable of explaining the rationale behind each prediction.

In the case of activity recognition, it may not be so relevant explaining the prediction of low level actions like, e.g., gestures based on inertial sensors patterns. The same is not true for high level activities like ADLs, since the prediction must take into account data from different sensors, temporal relationships between events, as well as low level actions. Important decisions in a smart living environment may rely on those predictions. Hence, inferring *why* the classifier predicted a particular ADL is a crucial step in providing solutions that are understandable, trusted and transparent. Consider, for example, an ADL recognition system that is deployed in the home of an elderly subject. The activities inferred by this system are continuously monitored by clinicians to support their diagnoses (e.g., cognitive decline). The context of execution of some activities may also be considered a behavioral anomaly that requires intervention. XAI would allow clinicians to trust the machine learning predictions, since explanations would make them more transparent and possibly also reveal relevant details. Explanations are also useful to refine the ADL recognition system by introducing, removing, or re-positioning sensors, as well as modifying algorithms and system parameters.

In this work, we propose a novel approach for eXplainable Activity Recognition (XAR) for ADLs performed in smart living spaces equipped with both wearable and environmental sensors. From each sensor data stream, we extract a stream of temporally characterised semantic states that, after preprocessing, is given as input to a trained classifier. For each ADL prediction, explanations are generated from a set of features over the semantic states, by considering both the feature importance and the feature values. Our solution is applicable to all classifiers from which a feature importance value can be derived from the model parameters.

In summary, the contributions of our work are the following:

- We introduce and formulate the problem of eXplainable Activity Recognition.
- We propose a data-driven explainable activity recognition method applicable to a large family of classifiers.
- We implemented the system and we provide a semantic-based metric to quantitatively evaluate the explanations on a real dataset of ADLs. By comparing two popular ML models, our results indicate the superiority of *one versus one* classifiers in providing explanations.

II. Related work

Explainable Artificial Intelligence (XAI) is emerging as an effective way to make machine learning processes more transparent [5]. Among other efforts, preliminary results from the participants of the XAI program launched by DARPA are available [4]. There are three main categories of XAI approaches: *interpretable models*, *model induction* methods (also called black box methods) and *deep explanation* based methods. Interpretable models are ML algorithms that are inherently explainable. For instance, Bayesian Rule Lists have been proposed to represent the ML model with probabilistic rules [6]. Well-known classifiers like decision trees are also inherently interpretable [7]. Model induction approaches consider the ML classifier as a black box [8]. Those methods analyze the input and the output of the classifier in order to reverse-engineer the rationale of the explanations. Finally, deep explanation methods are based on specifically designed deep neural networks that provide a sort of explanations [9]. Among these approaches we start our investigation on XAR with *interpretable models*. Indeed, it has been shown that activity recognition methods based on ad-hoc feature extraction and classic machine learning algorithms (e.g., Random Forests) are still competitive solutions with respect the ones based on deep learning [10].

There exists some preliminary work on explainable activity recognition. In [11], an interpretable classifier based on rule mining is proposed. The rules generated by the classifier are used as explanations. With respect to that work, our approach relies on standard machine learning algorithms commonly used for activity recognition. Other works that may be classified as XAR focus on the quite different task of activity recognition based on images, exploiting computer vision techniques [12]. In general, the majority of the existing solutions propose methods that generate complex association rules between input and output (even with interpretable models). We use machine learning models from which it is directly possible to compute feature importance, and we use a knowledge-based approach to combine the feature importance with the values of the features. Feature importance is well-known for being an human-understandable indication of the impact of each feature in classification [13].

Finally, while the generation of human readable explanations has been partially explored in the literature [14], this aspect was never investigated for sensor-based activity recognition.

III. Sensor-Based Explainable Activity Recognition

In this section, we describe our novel approach for real-time sensor-based eXplainable Activity Recognition (XAR). The overall pipeline of our method is depicted in Figure 1.

The user is continuously monitored through several environmental and/or wearable sensors that generate a stream of measurements. In the first step of our approach we derive, from each stream of raw sensors measurements, high-level semantic states. Intuitively, a semantic state describes what happens during a specific time interval (e.g., *the cooking stove was on from* t_1 *to* t_2). The semantic states obtained from each sensor are merged in a single stream. Then, we perform segmentation on the stream of semantic states and we apply a time-based feature extraction mechanism. A machine learning classifier is in charge of inferring the most likely activity given a feature vector. The goal of our method is to associate a set of semantic explanations to the prediction. For instance, suppose that the classifier predicted the activity ”Cooking”. An example of explanation could be “*the subject is likely standing in the kitchen, and she also just interacted with the stove*”.

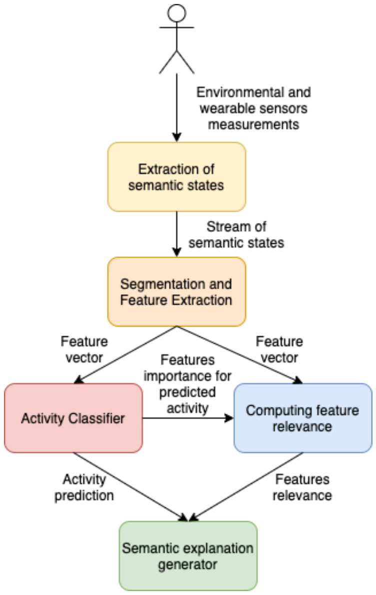


Fig. 1. The data flow of our eXplainable Activity Recognition method

For each feature, we extract the feature importance value from the parameters of the trained machine learning model. Then, by combining the feature importance value and the value of the feature in the feature vector, we compute the *feature relevance*: a value that indicates how relevant the feature was for the classification outcome. Note that the feature relevance value carries information related both to the input data and to the characteristics of the trained machine learning model. Finally, we generate semantic explanations by considering the predicted activity and the feature relevance values computed from the feature vector that lead to that prediction.

In the following, we describe our methodology in detail.

A. Extraction of semantic states

The types of sensing devices that we consider in our method belong to two categories: *binary sensors* and *value-based sensors*. We derive from each stream of raw sensor measurements a sequence of *semantic states*.

A binary sensor generates semantic states of two types corresponding to its two possible values, even if typically only one type is relevant for activity recognition. Consider, for instance, a magnetic sensor on the drawer used to store medicines. This sensor generates the state type *“Medicine_drawer_open”* whose temporal extent (the time the drawer remains open) may be relevant. The complementary state type *“Medicine_drawer_closed”* is less relevant since this is its *normal* state during all activities not involving medicines, and we ignore it.

On the other hand, value-based sensors can generate semantic states of several types. For example, an accelerometer that monitors user postures can generate semantic states of types *"Standing"*, *"Sitting"* and *"Lying"*.

In the following, we explain how we derive semantic states from binary and value-based sensors.

1) Semantic states generated by binary sensors: Binary sensors only generates the two values *ON* and *OFF* that can be easily mapped into semantic states. The most common binary sensors for activity recognition are environmental sensors. For example, when someone sits on the pressure mat sensor identified as P_2 on a kitchen chair, our method generates the semantic state *Using_kitchen_chair*$[t, -]$. Note that the end time of this semantic state is undefined (i.e., the state is currently *active*). Suppose that subsequently, at a time instant t', the same sensor P_2 generates a sensor event with value OFF. In this case our method, based on the knowledge that this measurement implies that the person that was sitting on the kitchen chair is now standing, updates the state to *Using_kitchen_chair*$[t, t']$.

Clearly we assume that for each binary sensor there should be a prior knowledge about the type of sensor, the corresponding object, the possible actions, and the sensor's location. We believe that this assumption is realistic, since this information can be gathered during the deployment of binary sensors.

2) Semantic states generated by value-based sensors: Mapping value-based sensor data to semantic states requires a more sophisticated approach. Consider, for instance, an accelerometer. This sensor continuously generates acceleration values on three axes at a high frequency. Clearly, it is not possible to directly associate semantics to those raw values.

Nonetheless, machine learning methods can be used to infer higher level information from the continuous stream of those inertial sensors. For instance, inertial sensor measurements generated by the sensors equipped on a wristband can be analyzed by machine learning classifiers to reliably derive simple gestures (e.g., the user is raising her arm, the arm is still, the user is performing some manipulation, etc). Deriving simple physical activities, postures and gestures from inertial sensors using machine learning is a well-established methodology in the literature.

Hence, in our methodology we apply machine learning algorithms to derive low-level activities from the streams of value-based sensors. Low-level activities are then mapped to semantic states. In particular, we generate a new semantic state when a user switches from a low-level activity a_1 to low-level activity a_2 For example, suppose that at time t_j, the user switched its low-level posture from standing to sitting and this is captured by a machine learning classifier from the inertial sensors data of the user's smartphone. Given this situation, our method updates the semantic state related to *standing* as $Standing[t_i, t_{j-1}]$ and generates a new semantic state related to *sitting* as $Sitting[t_j, -]$. Despite this low level classification is done with machine learning techniques, for the sake of this work we only consider simple patterns corresponding to low level actions that can be easily classified by existing techniques. Hence, we assume that explanations for these low level actions are not necessary.

B. Segmentation and feature extraction

In the following, we describe how we continuously segment the stream of semantic states to extract feature vectors. Our approach classifies the activity performed by the user each time the start or the end of a semantic state st is observed. We compute feature vectors that encode temporal dependencies between st and the semantic states that recently started or ended. Hence, given a state st that started or ended at time t, we build a segment $seg(st)$ that includes st and each semantic state whose time interval $[ts, te]$ has non-empty intersection with the time interval $[t', t]$, where t' is the time of occurrence of the K^{th} state update before t. Segments can have different durations, depending on the occurrence of the K updates observed before the end of the segment. Note that the underlying machine learning approaches that derive semantic states from value-based sensors (as described in Section III-A2) use a different segmentation strategy, relying on a window size that is significantly lower.

Given a segment of semantic states, we generate two different sets of features: *status based* features and *change point based* features. Status based features encode temporal dependencies between active and non-active semantic states. On the other hand, change point based features encode temporal dependencies between starting and ending times of semantic states. In *status based* features, for each state S that overlaps the segment interval, we compute a feature $fv^{ST}[S]$ that encodes the discounted sum of the time-based contributions considering an exponential temporal decay similarly to the method proposed in [15], with the difference that active states contribute with 1 to the discounted sum.

Change point based features are computed using a similar approach. For each status type S, we have two corresponding *change point* features: one related to the start time of state instances $f^{CP}[S^s]$ and one related to the end $f^{CP}[S^e]$.

The main difference with respect to status based features is that change point based features capture the temporal dependencies between the current semantic state with recent changes associated to other semantic states (distinguishing between beginnings and endings).

C. Activity classifier and feature importance

For each feature vector, a machine learning classifier outputs the probability distribution over the possible activities. The most likely activity is the predicted activity. In our methodology, the generation of explanations relies on the *feature importance* values that can be obtained from the parameters of the trained machine learning model. For each feature f_i, we derive from the model its importance I_{f_i}: a value that indicates the degree to which f_i discriminates the predicted activity with respect to the other activities. Each category of classifiers encodes feature importance values in a different way. For the sake of this work, we will focus on two machine learning classifiers that proved to be accurate for activity recognition and, at the same time, can be used in our study to obtain feature importance values in two significantly different ways: Random Forests and Linear SVM (i.e., SVM with linear kernel).

1) Feature importance in Random Forests: We extract feature importance values from Random Forests models by considering the Gini index values associated with each node of the forest. Intuitively, the Gini index indicates the probability of miss-classification given a condition on a specific feature. We extract feature importance by considering the mean decrease of the Gini index in order to derive which features can better discriminate activities [16]. Note that, independently from the method adopted to extract feature importance values in Random Forests, the value for each feature is independent from the activity predicted by the system because this model is inherently *multiclass*.

2) Feature importance in Linear SVM: Differently from Random Forests that are intrinsically *multiclass*, Linear SVM models are not. Indeed, we consider the *one versus one* classification strategy. Given two activity classes A_i and A_j, it is possible to understand which features discriminate A_i from A_j by analyzing the hyper-plane that separates them. A weight vector w_{ij} is associated with that hyperplane. The size of w_{ij} is the number of features and each element in w_{ij} indicates the importance of the corresponding feature for the binary discrimination between A_i and A_j. We extract feature importance values based on the activity predicted by the classifier. Suppose that the system predicts that the most likely activity is A_i. Hence, we can extract all the weight vectors w_{ij} that are associated to the hyper-planes that separate A_i with each other activity $A_j \in \mathbf{A}$ such that $A_i \neq A_j$. We aggregate those weight vectors by computing their weighted average, thus obtaining a single vector that encodes feature importance values. Given the hyper-plane that separates A_i and A_j, w_{ij} is weighted by $1 - p(A_j)$, where $p(A_j)$ is the probability of the activity A_j according to the classifier.

D. Feature relevance

Feature importance value reveals the impact of each feature in discriminating activities based on the trained model. However, feature importance alone is not sufficient to provide explanations, since the output of the classifier clearly depends on the specific values contained in the feature vector fv received as input. Intuitively, the most relevant features are the ones that are important according to the trained classifier and, at the same time, have high-values in the feature vector (considering the specific feature extraction mechanism proposed in Section III-B). On the other hand, a low value in the feature vector should decrease the relevance, and, similarly a low importance should also decrease the relevance.

More formally, given a feature f_i, the corresponding value $fv[f_i]$ in the feature vector fv, and its importance I_{f_i} according to the machine learning model, we obtain the *feature relevance* $fr(f_i, fv) = fv[f_i] \cdot I_{f_i}$.

E. Semantic explanations generator

Since feature importance and feature values include both model and input information and heavily influence the classifier in determining the predicted activity, we claim that feature relevance can be used as a basis to generate semantic explanations. We denote with $F^\star(fv)$ the set of features such that the relevance is greater than a threshold $\beta > 0$. The set $F^\star$ contains the most relevant features from which we want to generate the semantic explanations for the user. We generate a semantic explanation se for each feature $f_i \in F^\star$ based on the corresponding semantic state type S. As we explained in Section III-B, we consider status based features and change point based features.

Each status based feature f is associated with one state type S. If the value of this feature in the corresponding feature vector is greater than or equal to 1, it means that the state S is active. Hence, we show to the user an explanation like: *"The state S is currently active"*. Otherwise, if the value of the feature in the corresponding feature vector is lower than 1, we show an explanation like: *"The state S was recently observed"*. On the other hand, a change point based feature is associated with the begin or the end of a semantic state with type S. For each relevant change point based feature, we show to the user an explanation like: *"The state S recently started/ended to hold"*.

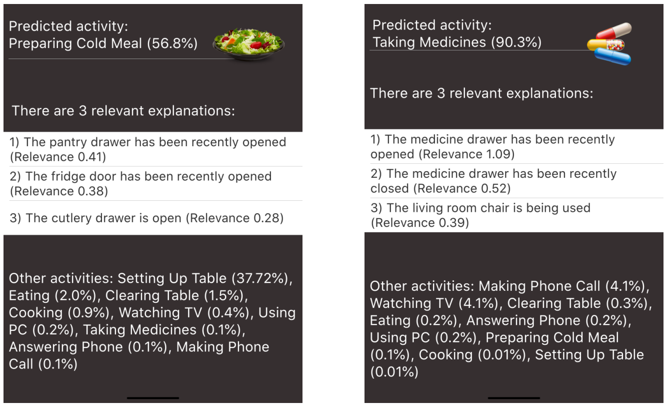
IV. Experimental Results

A. Dataset

In this work, we take advantage of a dataset of ADLs that we acquired in a controlled smart-home environment as part of a related research on activity recognition. For the sake of this evaluation, we focus on environmental sensor data only. The sensors considered in the experiments of this paper are: magnetic sensors that were positioned on some doors and drawers (e.g., fridge, medicine drawer, etcetera), pressure mat sensors on the chairs, smart-plug sensors to detect the usage of home appliances (e.g., electrical cooker, TV, etcetera), and virtual sensors on the mobile phones of the participants to understand when they make or receive a phone call. The activities that we consider are the following: *cooking*, *preparing a cold meal*, *setting up table*, *clearing table*, *eating*, *taking medicines*, *using PC*, *watching TV*, *making a phone call* and *answering phone*. The activities were performed by 12 subjects not involved in this research and instructed only about the organization of objects and tools in the smart-home and on the set of activities to be performed. The ground truth was acquired by cameras.

B. Examples of explanations

Before showing the actual evaluation, we provide some examples of explanations produced by the running prototype of our method. Figure 2 shows two specific examples, related to the activities *preparing cold meal* and *taking medicines*. Note that this interface maps the output of our method (see Section III-E) into more user-friendly sentences in natural language. In the *preparing cold meal* example, our interface shows the most relevant semantic states that involve the pantry drawer, the fridge and the cutlery drawer. These are likely interactions that occurred when users prepared cold meals in our dataset (e.g., preparing a salad). The *taking medicines*



Predicted activity:
Preparing Cold Meal (56.8%)

There are 3 relevant explanations:

1) The pantry drawer has been recently opened (Relevance 0.41)

2) The fridge door has been recently opened (Relevance 0.38)

3) The cutlery drawer is open (Relevance 0.28)

Other activities: Setting Up Table (37.72%), Eating (2.0%), Clearing Table (1.5%), Cooking (0.9%), Watching TV (0.4%), Using PC (0.2%), Taking Medicines (0.1%), Answering Phone (0.1%), Making Phone Call (0.1%)

(a) Preparing Cold Meal

Predicted activity:
Taking Medicines (90.3%)

There are 3 relevant explanations:

1) The medicine drawer has been recently opened (Relevance 1.09)

2) The medicine drawer has been recently closed (Relevance 0.52)

3) The living room chair is being used (Relevance 0.39)

Other activities: Making Phone Call (4.1%), Watching TV (4.1%), Clearing Table (0.3%), Eating (0.2%), Answering Phone (0.2%), Using PC (0.2%), Preparing Cold Meal (0.1%), Cooking (0.01%), Setting Up Table (0.01%)

(b) Taking medicines

Fig. 2. Examples of real explanations generated by our framework

example shows that the activity is explained by reporting that the user recently opened and closed the medicine drawer, but also by the fact that she is currently sitting on the living room chair (probably taking the medicines while sitting). Note that, in both examples, the first two explanations are related to recent changes in semantic states, while the last explanation is a currently active semantic state.

C. Metric: Explanation Score

In order to quantitatively evaluate the quality of the explanations generated by our approach, user-centric experiments should be carried out with Human Computer Interaction methodologies. However, during this work (mainly due to the Covid-19 pandemic) we were not able to recruit a sufficient number of users to conduct this study, and we plan it for future work. Nonetheless, we performed a quantitative assessment of our methodology by evaluating how much the explanations are consistent with respect to a common-sense knowledge about the relationships between activities and semantic states. We defined an OWL2 ontology that expresses those relationships. The ontology models, for each activity, its *partially explaining semantic states*. A semantic state partially explains an activity A if it explains (even if partially) A according to common-sense knowledge. For instance, in our ontology the semantic state *fridge opened* partially explains both the *cooking* and *taking medicines* activities, while it does not partially explains the activity *watching tv*. Our ontology also models groups of semantic states that partially explain activities. For instance, the semantic states *using kitchen chair* and *manipulating a fork* together partially explain the *eating* activity. Our ontology has been modeled by researchers of our group that were not aware of how the activities were performed in the reference dataset. We evaluate the degree at which our explanations can *partially explain* the predicted activity according to the ontology. Given a prediction A obtained from a feature vector fv, we compute the common-sense relevance $cr()$ of the explanation se_i corresponding to the feature f_i associated to a semantic state type S_i as follows:

$$cr(se_i, A) = \begin{cases} fr(f_i, fv) & \textit{if } S_i \textit{ partially explains } A \\ -fr(f_i, fv) & \textit{otherwise} \end{cases} \tag{1}$$

Hence, semantic states that do not partially explain the predicted activities are associated with a negative relevance, while the partially explaining ones are associated with a positive relevance. Then, we use the common-sense relevance to compute the *Explanation Score* (*Score* for the sake of brevity) that takes values in the range $[-1, 1]$:

$$Score(SE^\star, A) = \frac{\sum_{se \in SE^\star} cr(se, A)}{\sum_{se \in SE^\star} |cr(se, A)|} \tag{2}$$

where $SE^\star$ is the set of explanations that the system provides for the prediction of activity A. Explanations with a low relevance have a minor impact on the explanation score. If there are no explanations in $SE^\star$, the explanation score is -1. Note that, by using this formula, the features that do not explain A are associated with a negative common-sense relevance that penalizes the resulting score. We use this metric to perform a quantitative evaluation of the effectiveness of our approach.

D. Results

In our experiments, we performed leave-one-out cross validation to evaluate the generalization capability of our method in providing explanations to users that did not contribute to the training set with labeled data. At each fold, we train the classifier on 11 user and we test it on the remaining one. The test phase includes machine learning classification and semantic explanations generation. We use the F-1 score to measure the recognition rate. For each activity predicted by the classifier (independently if it is correct or not) we compute the *Explanation Score* using the metric in Equation 2. In order to provide a fair comparison with F1 score, we normalize the *Explanation Score* in the range $[0, 1]$. We repeated each experiment 1000 times and averaged the outcomes to show statistically robust results. We empirically determined the hyper-parameters (with grid search) as $K = 10$, $\chi = 0.8$, $\beta = 0.01$ for Random Forests, and $\beta = 0.25$ for Linear SVM. Figure 3 compares the *Explanation Score* and the F1 score for both Random Forests and Linear SVM. Since feature normalization is a common step in ML pipelines and it has a strong impact on feature values, we also show its impact on both classifiers.

While there are small differences in the recognition rate of Random Forests and Linear SVM, their explanation scores significantly differ. Indeed, Linear SVM reaches higher explanation scores. This is due to the fact that, since we considered a *one versus one* strategy for the Linear SVM classifier, we were able to extract feature importance values based on the predicted activity. On the other hand, feature importance in Random Forests does not depend on the predicted activity, and hence important features may be not relevant to explain every prediction. Normalization has almost no impact on the *Explanation Score* of Random Forests. This is probably due to

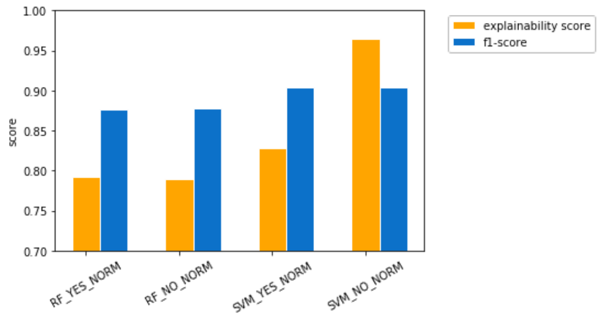


Fig. 3. Random Forests and Linear SVM with and without normalization

the fact that feature normalization has a small impact on the trained forest. Nonetheless, we observed that the *Explanation Score* on Linear SVM is significantly higher without normalization. We indeed observed that feature importance values of Linear SVM are significantly smoothed by normalization. Even though the classifier can still reach high F1 scores even with normalized features, the resulting feature importance values are significantly less informative. Figure 4 compares the F1 and the explanation scores for each activity for Linear SVM. We observed that *eating* and *setting up table* activities are poorly recognized by the classifier (due to an insufficient number of states that characterize it) while they are associated with a high explanation score. The few times that those activities are correctly recognized, they are associated to good explanations.

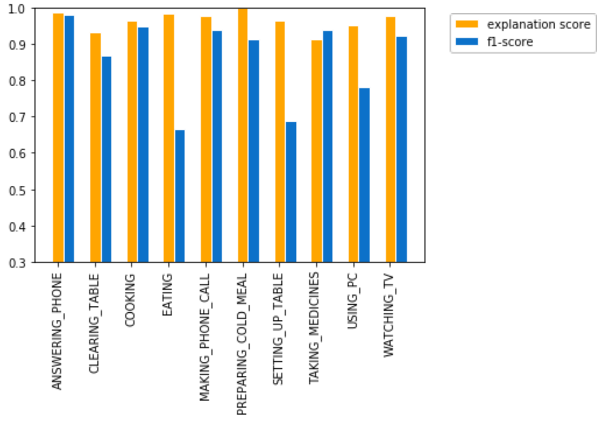


Fig. 4. Linear SVM: Explanation score vs F1-Score

V. Conclusion and Future Works

In this work, we proposed a novel approach for sensor-based eXplainable Activity Recognition specifically designed for activities of daily living. We believe that this work is a first important step into a promising research direction that we intend to further investigate in the following years. We indeed plan several related future investigations. A major limitation of our approach is that the semantic explanations do not show temporal dependencies between states. For instance, an explanation for *eating* could be "*the state manipulating fork started shortly after the start of the state sitting*". We will investigate how to extract such explanations from our semantic features that already encode some temporal dependencies. Finally, we aim at extending the evaluation of our framework with experiments on datasets including value-based sensors and by performing a case study based on HCI methodologies.

References

[1] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE T SYST MAN CY-S Part C*, vol. 42, no. 6, pp. 790–808, 2012.

[2] P. Rashidi and A. Mihailidis, "A survey on ambient-assisted living tools for older adults," *IEEE J. Biomed. Health*, vol. 17, no. 3, pp. 579–590, 2012.

[3] O. D. Lara, M. A. Labrador *et al.*, "A survey on human activity recognition using wearable sensors." *IEEE Comm. Surv. Tut.*, vol. 15, no. 3, pp. 1192–1209, 2013.

[4] D. Gunning and D. W. Aha, "Darpa's explainable artificial intelligence program," *AI Magazine*, vol. 40, no. 2, pp. 44–58, 2019.

[5] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[6] B. Letham, C. Rudin, T. H. McCormick, D. Madigan *et al.*, "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model," *Ann. Appl. Stat.*, vol. 9, no. 3, pp. 1350–1371, 2015.

[7] N. Narodytska, A. Ignatiev, F. Pereira, J. Marques-Silva, and I. RAS, "Learning optimal decision trees with sat." in *IJCAI*, 2018, pp. 1362–1368.

[8] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[9] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer Nature, 2019, vol. 11700.

[10] H. Gjoreski, J. Bizjak, M. Gjoreski, and M. Gams, "Comparing deep and classical machine learning methods for human activity recognition using wrist accelerometer," in *Proceedings of the IJCAI 2016 Workshop on Deep Learning for Artificial Intelligence, New York, NY, USA*, vol. 10, 2016.

[11] M. Atzmueller, N. Hayat, M. Trojahn, and D. Kroll, "Explicative human activity recognition using adaptive association rule-based classification," in *2018 IEEE International Conference on Future IoT Technologies (Future IoT)*. IEEE, 2018, pp. 1–6.

[12] M. Nourani, C. Roy, T. Rahman, E. D. Ragan, N. Ruozzi, and V. Gogate, "Don't explain without verifying veracity: An evaluation of explainable ai with video activity recognition," *arXiv preprint arXiv:2005.02335*, 2020.

[13] G. Casalicchio, C. Molnar, and B. Bischl, "Visualizing the feature importance for black box models," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 655–670.

[14] M. Bauer and S. Baldes, "An ontology-based interface for machine learning," in *Proceedings of the 10th international conference on Intelligent user interfaces*, 2005, pp. 314–316.

[15] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and mobile computing*, vol. 10, pp. 138–154, 2014.

[16] H. Han, X. Guo, and H. Yu, "Variable selection using mean decrease accuracy and mean decrease gini based on random forest," in *2016 7th ieee international conference on software engineering and service science (icsess)*. IEEE, 2016, pp. 219–224.